REMARKS

These remarks are set forth in response to the First Office Action. As this amendment has been timely filed within the three-month shortened statutory period, neither an extension of time nor a fee is required. At the time of the First Office Action, Claims 1 through 13 were pending and rejected in this application. Claims 1, 2, 6 and 10 are independent.

CLAIMS 1-13 ARE REJECTED UNDER 35 U.S.C. § 102(E) AS BEING ANTICIPATED BY U.S. PATENT APPLICATION PUBLICATION NO. 2006/0203980 TO STARKIE (HEREINAFTER STARKIE)

On pages 2-6 of the First Office Action, the Examiner asserted that Starkie discloses the invention corresponding to that claimed in Claims 1-13. This rejection is respectfully traversed.

The factual determination of anticipation under 35 U.S.C. § 102 requires the identical disclosure, either explicitly or inherently, of each element of a claimed invention in a single reference[1].1 Moreover, the anticipating prior art reference must describe the recited invention with sufficient clarity and detail to establish that the claimed limitations existed in the prior art and that such existence would be recognized by one having ordinary skill in the art[2]. As part of this analysis, the Examiner must (a) identify the elements of the claims, (b) determine the meaning of the elements in light of the specification and prosecution history, and (c) identify

[1] In re Rijckaert, 9 F.3d 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989); Perkin-Elmer Corp. v. Computervision Corp., 732 F.2d 888, 894, 221 USPQ 669, 673 (Fed. Cir. 1984).

[2] See In re Spada, 911 F.2d 705, 708, 15 USPQ 1655, 1657 (Fed. Cir. 1990); Diversitech Corp. v. Century Steps, Inc., 850 F.2d 675, 678, 7 USPQ2d 1315, 1317 (Fed. Cir. 1988).

corresponding elements disclosed in the allegedly anticipating reference.[3] This burden has not been met.

Claim 1

At the outset, Applicants note that no amendments have been made to any of Claims 1 through 13. Independent Claim 1 recites the following:

> A voice application simulation method comprising the steps of:
> loading a user simulation script programmed to specify simulated voice interactions with the voice application;
> deriving from the voice application a nominal output;
> generating a simulated output for the voice application corresponding to the nominal output; and
> conditionally producing a varying simulated input for the voice application.

On page 2 of the First Office Action, the Examiner relied upon paragraphs 0029, 0041-0051 to teach the limitations of Claim 1. Several of the passages are reproduced here for convenience.

> [0041] For speech recognition events, the name of the event is the same as the name of the corresponding grammar fragment, which comprises the name of the grammar being activated, and the top level nonterminal within that grammar, for example:
>
> [0042] "@airline::.Ask_timetablesform_destination_city", where "airline" is the name of the grammar and ".Ask_timetablesform_destination city" is the name of the top level nonterminal.
>
> [0043] For prompt events, the name of the event is the name of the prompt. Where a prompt is also described by a prompt grammar, the name of the prompt event is the name of the prompt grammar fragment, for example:
>
> [0044] "@airline.prompts::.Ask_timetablesform_city_of_departure"
>
> [0045] A number of names are reserved. For example, the name "@speaker::" is given to an event that represents the playing of prompts that are not represented by a prompt grammar. The names "@startcall::", "@endcall::", and

[3] 3 Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481 (Fed. Cir. 1984)

> "@hangup::" are given to the events where the dialog system answers a call, the dialog system ends a call, and when the speaker hangs up, respectively. Names of the form "@record::NAME-OF-VARIABLE" are given to events where a developer's response is recorded.
>
> [0046] Rather than storing a Markov model to describe the dialog, the scenario generator 206 includes the following three data structures or lists:
>
> [0047] (i) a list of all valid speech recognition events (Grammar Fragments), including a flag indicating whether the response to that event is expected to be the same regardless of the prompt played before it, and a count of the number of times that response has been given to the dialog system (i.e., the simulator 208);
>
> [0048] (ii) a list of all valid speech recognition events (Grammar Fragments) that can be spoken after each prompt is played; and
>
> [0049] (iii) a list of example interactions.
>
> [0050] The third structure provides the example interactions 234 that are subsequently output by the scenario generator 206, and is used to determine what responses the scenario generator 206 should give at particular points of the dialog.
>
> [0051] When generating menu driven example interactions, the scenario generator 206 operates in one of two submodes: a random walk mode, and a playback mode. The scenario generator 206 initially starts in random walk mode, with all three data structures empty. The VXAL state machine dialog 226 is executed by the simulator 208 and the output generated is stored in the list of example interactions.

As claimed, the voice application simulation method includes the steps of "deriving from the voice application a nominal output", and "generating a simulated output for the voice application corresponding to the nominal output". The cited passages of Starkie at paragraphs 0041-0051 fail to teach or suggest a voice application simulation method includes the steps of "deriving from the voice application a nominal output", and "generating a simulated output for the voice application corresponding to the nominal output". For example, the cited passages of Starkie at paragraphs 0041-0051 fail to discuss deriving a "nominal output" from the " voice application". The only deriving of a nominal output in the cited passages of Starkie is by the "scenario generator 206" not the voice application.

Moreover, the cited passages of Starkie at paragraphs 0041-0051 fail to discuss "generating a simulated output for the voice application corresponding to the nominal output". As the nominal output of the voice application is not even considered in the cited passages of Starkie, there cannot be, by definition, the generation of "a simulated output" that "correspond(s) to the nominal output."

Finally, the Examiner cites paragraph [0052] of Starkie as disclosing the claim limitation of "**conditionally producing a varying simulated input for the voice application**." Paragraph [0052] is reproduced here for convenience.

> [0052] The example interactions 234 are generated by a scenario generation process, as shown in FIG. 5. At step 502, the simulator 208 provides a question of the dialog to the scenario generator 206, together with a list of allowable responses or grammar fragments, along with the scopes (local or global) of those grammar fragments. The scenario generator 206 uses this information to update its lists of valid speech recognition events. A grammar fragment is classified as global if it is always active and can be spoken at any point. The scenario generator 206 initially assumes that the dialog system responds to global events in the same way, regardless of the question being asked, whereas the dialog system is expected to respond to local events in a different way depending upon the question being asked.

The cited passage of Starkie at paragraph 0052 fail to teach or suggest a voice application simulation method that includes the step of "conditionally producing a varying simulated input for the voice application". In reality, paragraph 0052 is silent with respect to any conditional production of a "varying simulated input for the voice application."

<u>Claims 2, 6 and 10</u>

Claims 2, 6 and 10 all recite the limitations of:

> generating a first simulated input for the voice application corresponding to a first pre-determined user input to the voice application, if the nominal output satisfies a first condition; and
>
> generating a second simulated input for the voice application corresponding to a second pre-determined user input to the voice application, if the nominal output satisfies a second condition different from the first condition.

On page 3 of the First Office Action, the Examiner relied upon paragraphs 0029, 0041-0055 to teach the limitations of Claims 2, 6 and 10. Several of the passages are reproduced here for convenience.

> [0053] At step 506, the scenario generator 206 randomly selects a grammar fragment that has not been considered in that dialog state, i.e., a fragment that is not represented in the first list with a non-zero count value. At step 508, a phrase is randomly generated using the selected grammar fragment and is passed to the simulator 208. At step 510, the list of used grammar fragments is updated to reflect the use of the selected fragment.
>
> [0054] These steps are repeated until all of the grammar fragments have been included in a phrase. If, at step 504, the scenario generator 206 cannot select an unused grammar fragment because all the available fragments have already been considered in that dialog state, the scenario generator 206 then examines the second list at step 518 to determine whether any dialog states exist in which not all available inputs have been considered. If no such states exist, then the third list is saved as example interactions 234 and the scenario generator 206 hangs up. Otherwise, if such states do exist, an unused state is selected at step 516 using the procedure described below, and the scenario generator 206 generates a preamble at step 514. The preamble is a sequence of events that moves the dialog from the state in which all inputs have been considered into the selected state in which not all inputs have been considered. Once the preamble has been executed or played back (at step 512) and the dialog is in the new state, the scenario generator 206 can resume its random walk.
>
> [0055] Rather than calculating a preamble from a Markov model, the scenario generator 206 uses a "lazy learning" technique to determine a preamble. A list of target states (T1) is determined from the scenario generator 206's two lists of valid speech recognition events. These states are states representing prompts

> in which not all inputs have been considered. Then, a list of desirable next inputs (NI) is generated from the two lists of valid speech recognition events. These inputs are valid inputs in that state in which the response of the dialog system is expected to be the same regardless of the previous prompt played.

As claimed, the voice application simulation method includes the steps of "generating a first simulated input for the voice application corresponding to a first pre-determined user input to the voice application, if the nominal output satisfies a first condition", and "generating a second simulated input for the voice application corresponding to a second pre-determined user input to the voice application, if the nominal output satisfies a second condition different from the first condition". The cited passages of Starkie at paragraphs 0041-0051 fail to teach or suggest a voice application simulation method that includes these two steps. For example, the cited passage of Starkie at paragraph 0053 teaches "the scenario generator 206 randomly selects a grammar fragment that has not been considered in that dialog state, i.e., a fragment that is not represented in the first list with a non-zero count value. At step 508, a phrase is randomly generated using the selected grammar fragment and is passed to the simulator 208."

Accordingly, the scenario generator 206 of Starkie does **NOT** "generat(e) a first simulated input for the voice application corresponding to a first pre-determined user input to the voice application, if the nominal output satisfies a first condition. Nor does Starkie generat(e) a second simulated input for the voice application corresponding to a second pre-determined user input to the voice application, if the nominal output satisfies a second condition different from the first condition."

Thus, for the above-described reasons, the Examiner has failed to establish that Starkie identically discloses the claimed invention, as recited in independent Claims 1, 2, 6 and 10, within the meaning of 35 U.S.C. § 102. Applicants, therefore, respectfully submit that the imposed rejection of Claims 1-13 under U.S.C. § 102 for anticipation based on Starkie is not factually viable and, hence, solicit withdrawal thereof.

For these reasons, the Applicants respectfully request the withdrawal of the rejections under 35 U.S.C. § 102(e). This entire application is now believed to be in condition for allowance and such action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: November 23, 2007

/Steven M. Greenberg/

Steven M. Greenberg
Reg. No.: 44,725
Customer Number 46322
Attorney for Applicant(s)
Carey, Rodriguez, Greenberg & Paul, LLP
950 Peninsula Corporate Circle, Suite 3020
Boca Raton, FL 33487
Tel: (561) 922-3845
Fax: (561) 244-1062